

# DUGGA: Objektorienterade applikationer

## Läs detta!

- *Uppgifterna är inte avsiktligt ordnade efter svårighetsgrad.*
- Börja varje uppgift på ett nytt blad.
- Skriv ditt namn och personnummer på varje blad (så att vi inte slarvar bort dem).
- **Skriv rent dina svar. Oläsliga svar rä t t a s e j!**
- Programmen skall skrivas i Java 5 eller senare, vara indenterade, renskrivna och i övrigt utformade enligt de principer som lärts ut i kursen.
- Onödigt komplicerade lösningar ger poängavdrag.
- Programkod som finns i tentamenstesen behöver ej upprepas.
- Givna deklARATIONER, parameterlistor, etc. får ej ändras, såvida inte annat sägs i uppgiften.
- Läs igenom tentamenstesen och förbered ev. frågor.

I en uppgift som består av flera delar får du använda dig av funktioner klasser etc. från tidigare deluppgifter, även om du inte löst dessa.

*Lösningarna fylls i direkt i tesen!*

*Lycka till!*

**Uppgift 1 Modellering**

Rita ett klassdiagram som bäst överensstämmer med följande relationer mellan klasserna A-F:

- En A har två B
- Flera C känner till A
- B använder D
- E är en D och implementerar F

För full poäng krävs att pilarna är av rätt typ och riktning.

(3 p)

## Uppgift 2 Gui

I en viss applikation med grafiskt användargränssnitt vill man kunna välja mellan olika valutor när en prisuppgift visas. I följande exempel visas priset alltid i svenska kronor (SEK).

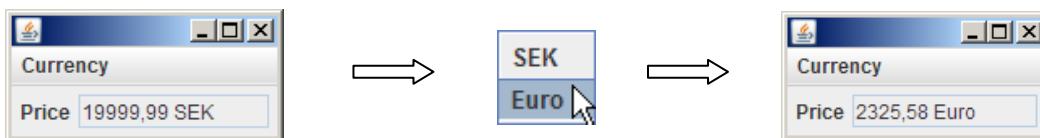
```
public class MainWindow extends JFrame {
    public static final double SEK2EURO = 1/EURO2SEK;
    private double price = 19999.99; //SEK
    private JTextField priceView;

    public MainWindow() {
        setLayout(new FlowLayout());
        add(new JLabel("Price"));
        priceView = new JTextField(10);
        priceView.setText(format(price, "SEK"));
        priceView.setEditable(false);
        add(priceView);
        pack();
        setVisible(true);
    }

    // Plats för lösningen finns på nästa sida -->

    // Format a double into a string with two decimals
    private String format(double price, String currency) {
        return String.format("%.2f %s", price, currency);
    }
}
```

Utvidga programmet med en meny "Currency" så att användaren kan välja i vilken valuta priset visas. Man skall kunna välja mellan alternativen SEK och Euro. Exempel på hur det skall se ut när användaren i menyn väljer att visa priset i Euro:



Skriv en metod som skapar menyn. Det räcker sedan om du implementerar menyalternativet för Euro.

(4 p)

*uppg. 2 forts.*

### Uppgift 3    Observer

Vid användning av Observer-mönstret kan man ha både flera observerade objekt och flera observatörer. T.ex. kan en observatör observera två olika objekt. Antag att vi har klasserna

```
public class A extends Observable { ... }  
public class B extends Observable { ... }
```

Antag att både A och B kan anropa `notifyObservers` med en sträng *eller* ett heltal som argument.

```
public class Obs implements Observer {  
    public Obs(A a,B b) {  
        a.addObserver(this);  
        b.addObserver(this);  
    }  
    public void update(Observable o, Object arg) {
```

```
        }  
    }
```

Metoden `update` i `Obs` skall fungera så här: Om anropet orsakades av att A anropade `notifyObservers` med en sträng som argument så skall strängen översättas till stora bokstäver och skrivas ut. Om istället anropet orsakades av att B anropade `notifyObservers` med ett heltal som argument så skall talet multiplicerat med 10 skrivas ut. Skriv färdigt `update`!

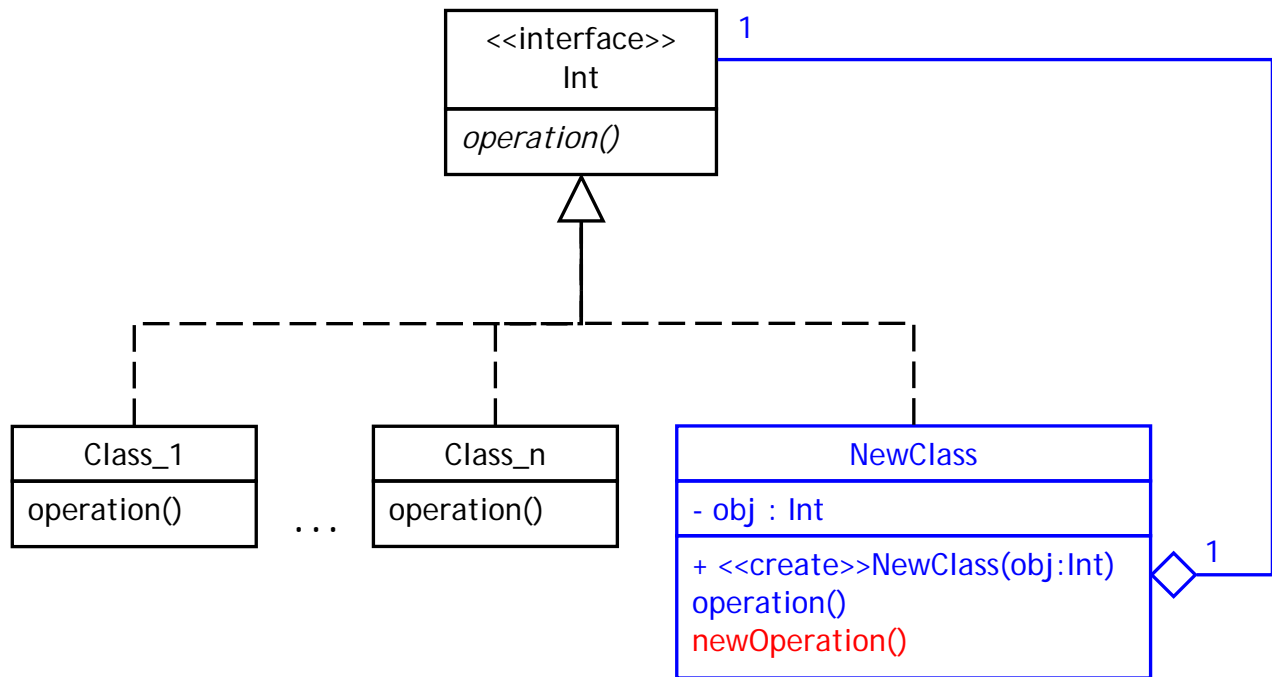
*Anm.* A- och B-objekten kan inte nås via instansvariabler.

(3 p)

#### Uppgift 4 Designmönster

a)

Vilket designmönster beskrivs schematiskt av följande diagram?



(1 p)

b)

Beskriv i javakod hur designmönstret Singleton realiseras.

(3 p)

### Uppgift 5 Strömmar och filer

Antag att man har en binär datafil av heltal. Första talet i filen är av typen `int` och det anger antalet efterföljande tal som alla är av typen `long`. Problemet är att läsa en sådan fil och skriva ut talen, utom det första av typ `int`, som textsträngar till en textfil, en per rad. (Felhantering med undantagshanterare kan utelämnas.)

a)

Vilka av följande strömklasser bör användas för läsning av binärfilen, resp. skrivning av textfilen: `FileWriter`, `ObjectInputStream`, `FileReader`, `DataOutputStream`, `DataStream`, `FileInputStream`, `InputStream`, `PrintWriter`, `ObjectOutputStream`, `DataInputStream`? Ange hur strömvariablerna deklarerats samt hur strömobjekten skapas. Kalla variablerna `in` resp. `out`. Filnamnen är ”`infile.dat`” och ”`outfile.txt`”.

(2 p)

b)

konstruera en loop som läser talen från datafilen och skriver motsvarande siffersträngar till textfilen.

(3 p)

## Uppgift 6 Trådar

Antag att du har funktioner som av någon anledning tar lång tid att beräkna. För enkelhets skull antar vi att dessa funktioner tar ett flyttal som parameter och ger ett flyttal som resultat. Det skulle vara bra om en eller flera sådana beräkningar kunde utföras i bakgrunden i ett program, samtidigt som programmet fortsätter med andra aktiviteter, t.ex. att underhålla användaren medan denne väntar på resultaten. Ett sätt att åstadkomma bakgrundsberäkningar är följande. Definiera först gränssnittet

```
public interface Computable {  
    float compute(float arg);  
}
```

Funktionen som utför den tidskrävande beräkningen placeras sedan i en subclass till `Computable`, t.ex.

```
public class SlowFox implements Computable {  
    float compute(float arg) { ... }  
}
```

Man definierar sedan klassen `DoBackground`. Ett objekt av klassen skall exekvera `compute`-metoden i ett `Computable`-objekt i en egen tråd. Så här skall det fungera. Antag att du vill exekvera `SlowFox.compute` med argumentet `123.8`

```
DoBackground job = new DoBackground(new SlowFox(), 123.8);  
  
// do some other work here ...  
  
job.join();  
float x = job.getValue();
```

Konstruktorn skall ta `compute`-metodens inparameter som argument och värdet i objektet. Resultatet från `compute` skall sparas i en instansvariabel som senare kan läsas av med metoden `getValue`. Om `getValue` anropas innan resultatet är klart skall undantaget `IllegalStateException` kastas. Använd en boolesk instansvariabel för att hålla reda på om beräkningen är klar eller ej. Implementera `DoBackground`. Plats finns på nästa sida



```
public class DoBackground
{
    // put instance variables here
```

```
    public DoBackground(Comparable obj, float arg) {
```

```
    }
```

```
    public float getValue()
    {
```

```
    }
```

```
}
```

(4 p)

### Uppgift 7    Kommunikation

Komplettera nedanstående program så att det upprepade gånger läser textrader från tangentbordet och sänder varje rad som ett datagram till mottagardatorn vars adress och portnummer ges som programargument vid exekveringen.

```
public class Datagramsender {  
    public static void main(String[] arg) {  
        String hostAddress = arg[0];  
        int port = Integer.parseInt(arg[1]);
```

```
    }  
}
```

(4 p)

### Uppgift 8 Reflektion

Antag att klassen C implementerar gränssnittet

```
public interface Int {  
    void f();  
}
```

Skriv ett kodavsnitt som laddar klassen c, skapar ett objekt av klassen, samt anropar metoden f.

(3 p)